

# Implementarea filtrelor digitale IIR în forma lattice

Laborator 8, PSS

## Table of contents

1	Obiectiv	1
2	Noțiuni teoretice	1
3	Exerciții teoretice	2
4	Exerciții practice	3
5	Întrebări finale	4

## 1 Obiectiv

Familiarizarea studenților cu formele de implementare tip *lattice* folosite la implementarea filtrelor de tip IIR

## 2 Noțiuni teoretice

Implementarea în formă *lattice-ladder* a unui filtru IIR de ordin 3:

Ecuatii:

$$H(z) = \frac{C(z)}{A(z)}$$

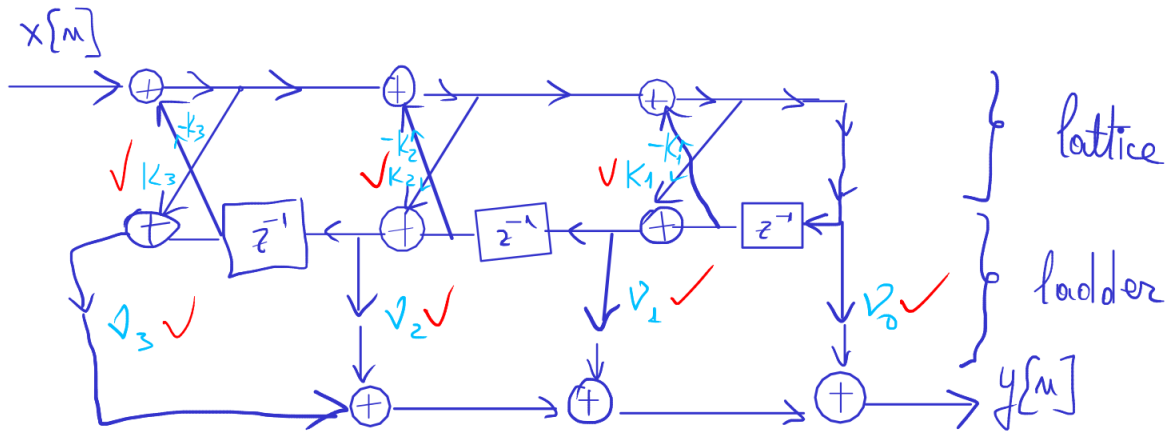


Figure 1: Forma lattice-ladder, IIR ordin 3

Coefficienții de reflexie  $K_i$  se găsesc exact ca la sistemele FIR (lab. precedent):

$$\begin{aligned}
 A_0(z) &= B_0(z) = 1 \\
 A_m(z) &= A_{m-1}(z) + K_m \cdot z^{-1} \cdot B_{m-1}(z) \\
 A_{m-1}(z) &= \frac{A_m(z) - K_m \cdot B_m(z)}{1 - K_m^2} \\
 B_m(z) &= z^{-m} B_m(z^{-1}) = \text{similar cu } A_m(z), \text{ cu coeficienții în ordine inversă}
 \end{aligned}$$

Suplimentar, pentru coeficienții  $\nu_i$  se folosește o ecuație asemănătoare:

$$C_{m-1}(z) = C_m(z) - \nu_m \cdot B_m(z)$$

### 3 Exerciții teoretice

1. Fie sistemul IIR cauzal cu poli și zerouri, cu funcția de sistem:

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2} + 2z^{-3}}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Determinați și desenați structura echivalentă *lattice* cu poli și zerouri.

2. Se dă sistemul IIR cauzal numai cu poli, cu funcția de sistem:

$$H(z) = \frac{1}{1 + \frac{2}{5}z^{-1} + \frac{7}{20}z^{-2} + \frac{1}{2}z^{-3}}$$

Determinați coeficienții structurii *lattice* și desenați-o.

## 4 Exerciții practice

1. În Matlab, utilizați utilitarul `fdatool` pentru a proiecta unul din filtrele următoare:
  - a. Un filtru trece-jos IIR de ordin 4, de tip eliptic, cu frecvența de tăiere de 6kHz la o frecvență de eșantionare de 44.1kHz;
  - b. Un filtru trece-sus IIR de ordin 4, de tip eliptic, cu frecvența de tăiere de 2.5kHz la o frecvență de eșantionare de 44.1kHz;
  - c. Un filtru trece-bandă IIR de ordin 4, de tip eliptic, cu banda de trecere între 0.5kHz și 5.5kHz la o frecvență de eșantionare de 44.1kHz.

2. Exportați coeficienții filtrului de mai sus în Workspace-ul Matlab, cu numele `Num` și `Den`.

Utilizați funcția `tf2latc()` pentru a converti coeficienții din vectorii `Num` și `Den` (corespunzători formelor directe) în coeficienții `K` și `V` ai formei *lattice*.

Utilizați apoi și funcția inversă `latc2tf()` pentru a converti coeficienții din forma *lattice* `K` și `V` înapoi în coeficienți ai formei directe, și verificați că se obțin aceleași valori ca în `Num` și `Den`.

### Observația 1:

- Utilizând `latc2tf()` nu veți obține exact coeficienții inițiali din `Num`, ci coeficienții raportați la prima valoare, `Num / Num(1)`. Acest lucru se întâmplă pentru că forma de implementare *lattice* necesită ca primul coeficient din  $H(z)$  să fie neapărat de valoare 1.

Așadar, coeficienții din vectorul `K` nu implementează de fapt filtrul `Num`, ci filtrul `Num / Num(1)`. Asta înseamnă că în schema Simulink trebuie ca semnalul de intrare să fie trecut printr-un Gain suplimentar înainte de a intra în schema *lattice*, de valoare `Num(1)` (adică  $b_0$ ).

3. În mediul Simulink, realizați implementarea IIR filtrului de mai sus în forma *lattice*.
  - a. Implementați schema, punând în Gain-uri coeficienții din vectorii `K` (`K(1)`, `K(2)`, etc) și `V` (`V(1)`, `V(2)`, etc)
  - b. Puneți la ieșire un bloc de vizualizare ("Scope") și afișați răspunsul la impuls (intrarea "Discrete Impulse") și răspunsul la treapta unitate (intrarea "Step").
4. În Matlab, creați o funcție pentru a filtra un vector `x` cu un filtru IIR pentru care se cunosc coeficienții *lattice* `K` și `V`, obținând vectorul de ieșire `y`.

Găsiți o metodă de a calcula ieșirea `y[n]` la un moment oarecare `n`, pe baza schemei. Cu alte cuvinte, dacă avem schema, cum putem scrie niște ecuații sau linii de cod pentru a o implementa?

Puteți utiliza următorul template orientativ:

```

function y = filter_lattice_IIR(K, V, x)
% Filter the vector x with a lattice IIR filter with coefficients K and V

ord = length(K);

% Write code here:

for i=1:length(x)
    y(n) = ...
end

end

```

5. Testați funcția, apelând-o cu coeficienții K și V ai filtrului proiectat anterior, pentru semnalul de intrare **x** de forma:

- a. un vector cu un 1 urmat de 19 zerouri (impuls unitate)
- b. un vector cu 20 de 1 (treapta unitate)

Afișați grafic vectorii obținuți la ieșire, cu funcția **stem()**.

## 5 Întrebări finale

1. TBD